NPS-62-87-001

NAVAL POSTGRADUATE SCHOOL
Monterey, California

MODEL-BASED BUILDING VERIFICATION
IN AERIAL PHOTOGRAPHS

by

Chin-Hwa Lee

September 1987

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943

Rear Admiral R. C. Austin   D. A. Schrady
Superintendent   Provost

The work reported herein was supported in part with funds
provided by the Joint Directors of Laboratories.

Reproduction of all or part of this report is authorized.

This report was prepared by:

Chin-Hwa Lee
Associate Professor

Reviewed by: Released by:

John P. Powers   Gordon E. Schacher
Chairman   Dean of Science and
Electrical and Computer Engineering   Engineering

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS-62-37-001	2. GOVT ACCESSION NO *H: /*	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Model-Based Building Verification in Aerial Photographs		5. TYPE OF REPORT & PERIOD COVERED Annual Report for Period October 1985-September 1986
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Chin-Hwa Lee		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943-5100		10. PROGRAM ELEMENT PROJECT TASK AREA & WORK UNIT NUMBERS PE63701B, 3208, 076 MIPR # HM0050-6-357
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943-5100		12. REPORT DATE 21 October 1986
		13. NUMBER OF PAGES 34
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Cmdr. Robert Booker Defense Mapping Agency Headquarter Bldg 56c, U.S. Naval Observatory Washington, D.C.		15. SECURITY CLASS (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Parallel Architecture, Content Addressable Memory, Object Detection, Image Correlation		

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper concerns the design of a computer vision system for change detection. Here, change detection is defined as figuring out the differences between an object model and the newly sensed image. The target objects are confined to the cultural features, such as roads and buildings. We divide the task into two modules: model verification and image interpretation. In this report, the verification stage will be discussed in detail.

DD _{1 JAN 73} FORM 1473   EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

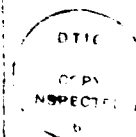# Model-Based Building Verification
in Aerial Photographs

Hsi-Jian Lee

Chin-Hwa Lee

Department of Electrical and Computer Engineering

Naval Postgraduate School

Monterey, CA 93943

U. S. A.

May 1986

A-1

I. INTRODUCTION

This paper concerns the design of a computer vision system for change detection. Here, change detection is defined as figuring out the differences between an object model and the newly sensed image. The target objects are confined to the cultural features, such as roads and buildings. We divide the task into two modules: model verification and image interpretation. In this report, the verification stage will be discussed in detail. In general there exists a lot of domain specific heuristics to judge the status of changes. For example, to verify the existence of a building, we can check its shape, size, height, surface direction, and surface material, etc. The expert system approach is a natural approach which can code all information together. While different photo interpreters and field specialists may have different viewpoints about the status of an object, expert systems can be modified easily to reflect a particular one's viewpoint.

Verification of existent object, i.e., modeled objects, is the first module of our change detection system. Although it is simpler than full interpolation, it has a lot of applications. The final goal will be the interpretation of the objects which are concluded 'changed' in the first stage. This, however, needs more heuristic interpretation rules. Under current design methodology, we need only append these heuristics to the knowledge base and add new interpretation control rules. The verification subsystem structure will not be affected or, more conservatively, will be affected only to a minimal extent.

Brooks [1.2] proposed a domain-independent model-based computer vision system, ACRONYM. The system uses a volumetric primitive, generalized cone, to describe generic objects. The parameters can be expressed not only by a specific numeric value but also by a set of mathematic expressions. The modelled objects can thus have variations in size and structure. The spatial relationship between two affixed parts is represented by coordinate transformation, including a translation and a rotation. By changing the transformation, variations in object spatial relationship can be modelled. Through an algebraic manipulation system and a geometric manipulation system, invariant and quasi-invariant features can be predicted. Edge- based, goal-guided image segments are then matched with the prediction, and image interpretations are then given. Although the system is flexible enough, it does not give enough examples to show the general applicability. We believe that a specific vision system is usually applicable to a special case where, for example, the objects may be restricted or the camera model is precisely known.

Nazif and Levine [3] designed a rule-based system to segment an image, a low-level image processing task. Traditionally, there are two disciplines for image segmentation. One is a region approach, based on locating homogenous regions; another is an edge approach, based on locating the gray scale discontinuities in the image. Each approach has its merits and shortcomings. Through an expert system, different cues can be coded into rules for image segmentation. For example, an edge cue can be used to split a region or to merge two (or more) adjacent regions. Region cue can also be used to join two lines. In addition to these knowledge rules, the system used metarules to control the operations of

the knowledge rule. Focus of attention rules are also incorporated to determine the path of processing within the images.

McKeown et al. [4] designed a rule-based system, SPAM, for the interpretation of airport imagery based on a world model. They used region-based algorithm to segment images. Then region properties such as shape, texture, spectral properties, etc., are extracted in order to determine the classes of airport features of the region. Multiple fragments, or say, classes, may be assigned as the interpretation result of a single region. The fragments with close physical proximity and often related function are organized into a functional area. After the functional areas are formed, the mutually consistent ones are used to represent an airport in the verification process.

Barrett et al. [5] proposes an automatic symbolic change detection (ASCD) system. It is a knowledge based system which looks in the reference data set for features of interest and then processes and attempts to identify the corresponding features in the mission data set. They spent a significant amount of effort in developing the knowledge base and the rules for the identification of the features, including six topographic and six hydro- graphic features.

Tavakoli and Rosenfeld [6] describe a procedure for the recognition of cultural features such as buildings and roads on aerial photographs. They use an edge-based method to interpret the posssible features. Straight line segments are fitted to a set of edge pixels. Based on gray level and geometric information, segments are grouped into road-like and building-

like groups.

Price and Reddy [7] developed a system for symbolic registration and change analysis, determining what changes in feature values occurred between two views of the scene, finding the corresponding regions in the two images. They apply these techniques to compare the pair of images to generate descriptions of the changes in the scene. In order to improve the system performance, all analyses are performed symbolically. The changes they sought include the scale difference, the translation difference, and the sun angle difference.

II. SYSTEM OVERVIEW

The proposed system structure for change detection is shown in Fig. 2.1. This phase can be divided into four modules. Module A is for image pre- processing. It includes optional smoothing, segmentation and feature extraction. Module B is for knowledge retrieval. Given the sensed image, we can extract part of the environment model as the verification basis. That is, the range of the input images is given in geodetic coordinate system, and we can use this value to select a block from the input image. Module C is the knowledge-based verification subsystem. This is the main topic of this paper. Inputs are the image and the extracted models. They are checked in a depth-first tree traversal manner. Output will be the status of the known objects in the input image. The changed objects will be fitted into the next module. Module D is the knowledge-based interpretation system.

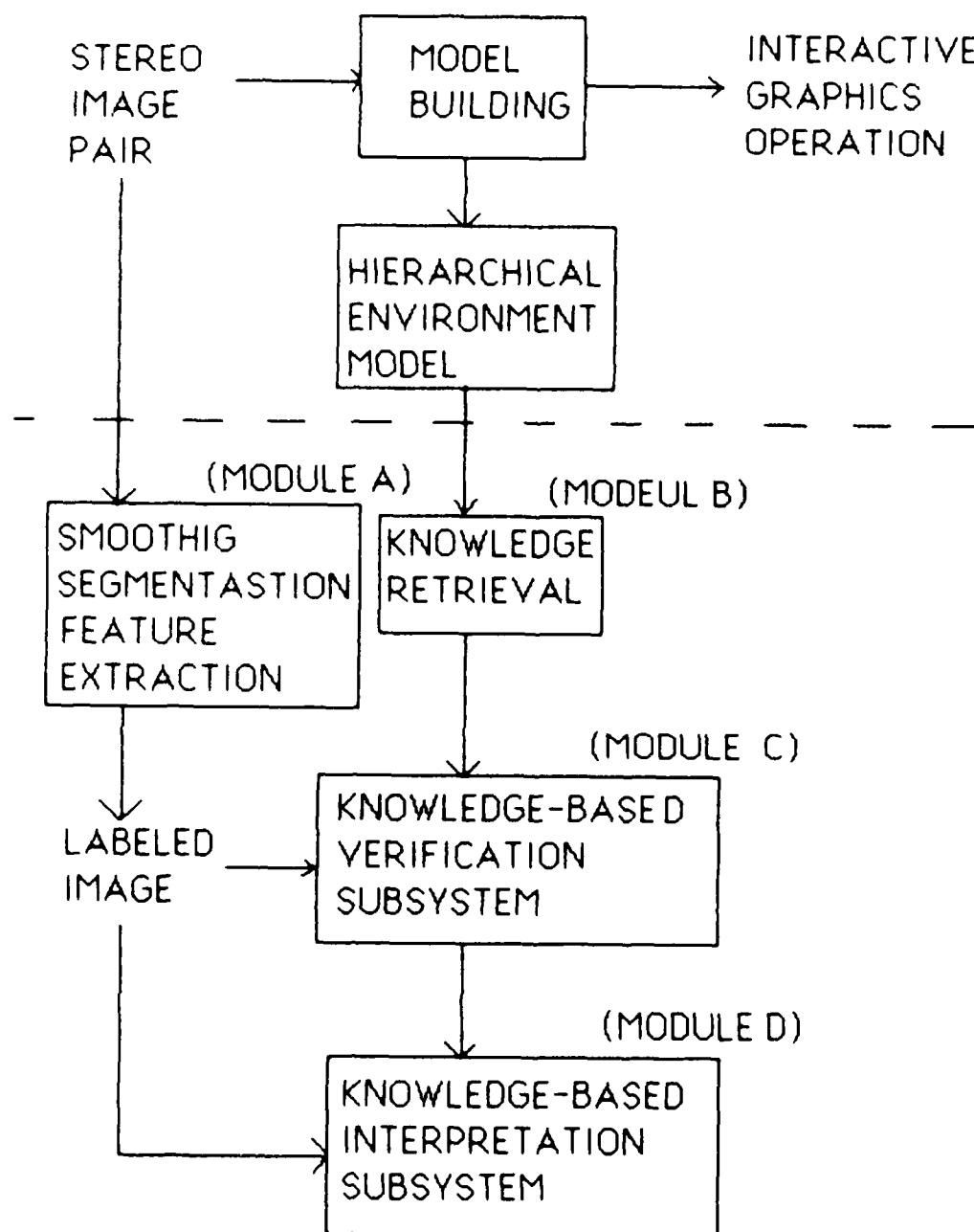Fig. 2.2 shows the control flow of the proposed interpretation sub

FIG 2.1 DIAGRAM OF A CHANGE DETECTION SYSTEM

system. Input to the subsystem will be the unverified objects and un-interpreted regions. From verified objects and the generic world model the system will predict possible interpretations, and the interpretation module will gather evidences to verify them. The results, that is the new identification of unverified objects and the uninterpreted regions, are finally used to update the model.

In the system diagram of Fig. 2.2 we also propose a model-guided correspondence solver. When a new interpretation is predicted evidences are needed. We can collect this evidence from the stereo image. Using this solver, we can obtain the 3-D information by using the same method that we used to build the hierarchical environmental model. However, the correspondence problem is nontrivial. This difficulty is experienced during the finding the features. We feel that enough knowledge is required for a good correspondence solver. A knowledge-based disparity analysis system is being developed.

2.1 Knowledge Model Creation

This section concerns the creation of an environmental model. Our proposed verification subsystem depends heavily on the environmental model. We have developed a simple, interactive method to create the 3-D object model. From the stereo principle, we know that if the correspondence problem is solved, we can find the object coordinates, (X, Y, Z), through the triangulation method.

The perspective projection of a 3-D object point (X, Y, Z) into a

STEREO
IMAGES

UNVERIFIED
REGIONS

VERIFIED
OBJECT

```
┌─────────────────┐        ┌──────────────┐   KNOWLEDGE
│ MODEL-GUIDED    │        │              │ ← SOURCE
│ CORRESPONDENCE  │        │ PREDICTION   │   GENERIC
│ SOLVER          │        │              │   MODEL
└─────────────────┘        └──────────────┘
```

STRUCTURE
INFORMATION

```
┌──────────────────────────┐
│ INTERPRETATION           │
│ EVIDENCE GATHERING       │
└──────────────────────────┘
```
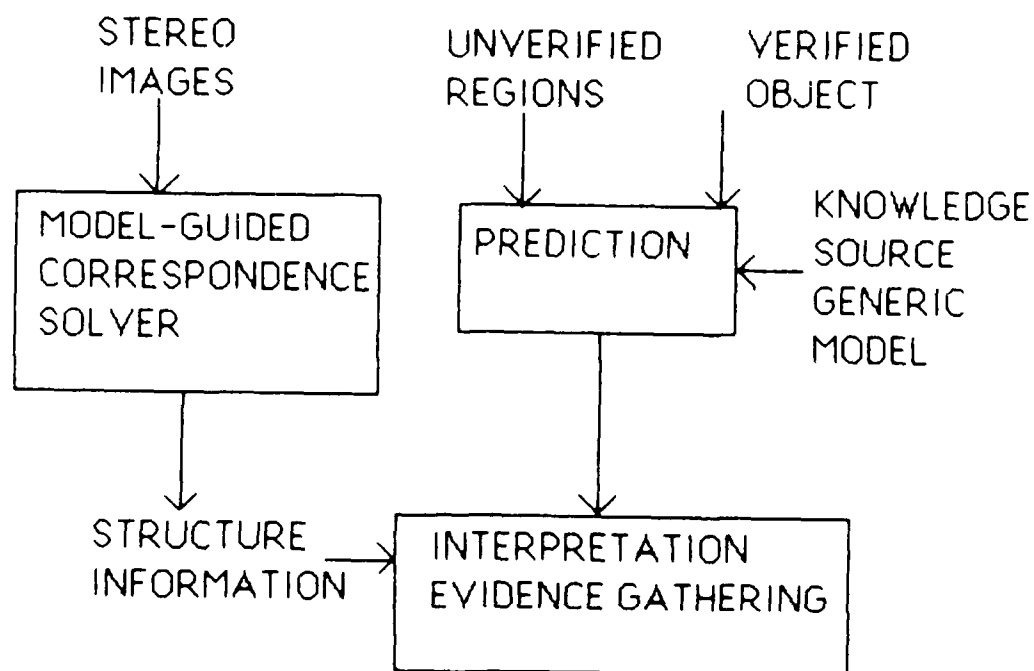
FIG.2.2 CONTROL FLOW OF THE INETERPRETATION
SUBSYSTEM

8

2-D image coordinate system. can be expressed as [Ref]

$$\begin{pmatrix} x & - & x_0 \\ y & - & y_0 \\ & -f & \end{pmatrix} = k * \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} * \begin{pmatrix} X & - & X_L \\ Y & - & Y_L \\ Z & - & Z_L \end{pmatrix} \qquad (1)$$

where

(1) (x, y) and (X, Y, Z) are the 2-D and 3-D image point coordinate. (x_0, y_0) are the principal point of the camera. +

(2) k is a scaling factor.

(3) $m_{ij}, i = 1, 2, 3$ and $j = 1, 2, 3$ are the components of the rotation matrix.

(4) f denotes focal length. Eq. (1) can be rearranged and put in the following form:

$$X = a + v * s$$
$$Y = b + v * t$$
$$Z = c + v * u$$

Where,

$$a = X_L$$

$$b = Y_L$$

$$c = Z_L$$

$$v = 1/k$$

$$s = m_{11}(x - x_0) + m_{21}(y - y_0) - m_{31}f$$

$$t = m_{12}(x - x_0) + m_{22}(y - y_o) - m_{32}f$$

$$u = m_{13}(x - x_0) + m_{23}(y - y_0) - m_{33}f$$

9

These values are known. However, (X, Y, Z) and v are unknown variables, so we can use left and right images to obtain two set of a, b, c, v, s, t, and u. Put into matrix form, we get

$$\begin{bmatrix} 1 & 0 & 0 & -S_1 & 0 \\ 0 & 1 & 0 & -t_1 & 0 \\ 0 & 0 & 1 & -u_1 & 0 \\ 1 & 0 & 0 & 0 & -s_2 \\ 0 & 1 & 0 & 0 & -t_2 \\ 0 & 0 & 1 & 0 & -u_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \end{bmatrix}$$

Here, subscripts 1 and 2 denote left and right images, respectively. In short form, we have

$$A \cdot U = B \qquad (4)$$

It includes six equations and 5 unknowns, (X, Y, Z, v_1, v_2). Solving by pseudo-inverse method, the solution can be expressed as

$$U = (A^T A)^{-1} \cdot A^T \cdot B \qquad (5)$$

In geometry, the 3-D position of a point, (X, Y, Z), is the intersection point of two straight lines, which are the lines passing through the left and right image coordinate and the respective camera center. If the distance between two cameras is not large enough, both lines will be nearly parallel. The solution (X, Y, Z) will be very sensitive with respect to the noise perturbation of the point.

2.2 System Development Environment

The current system development is shown in Fig. 2.3. Basically, we implement the whole system under a general purpose computer sytem.
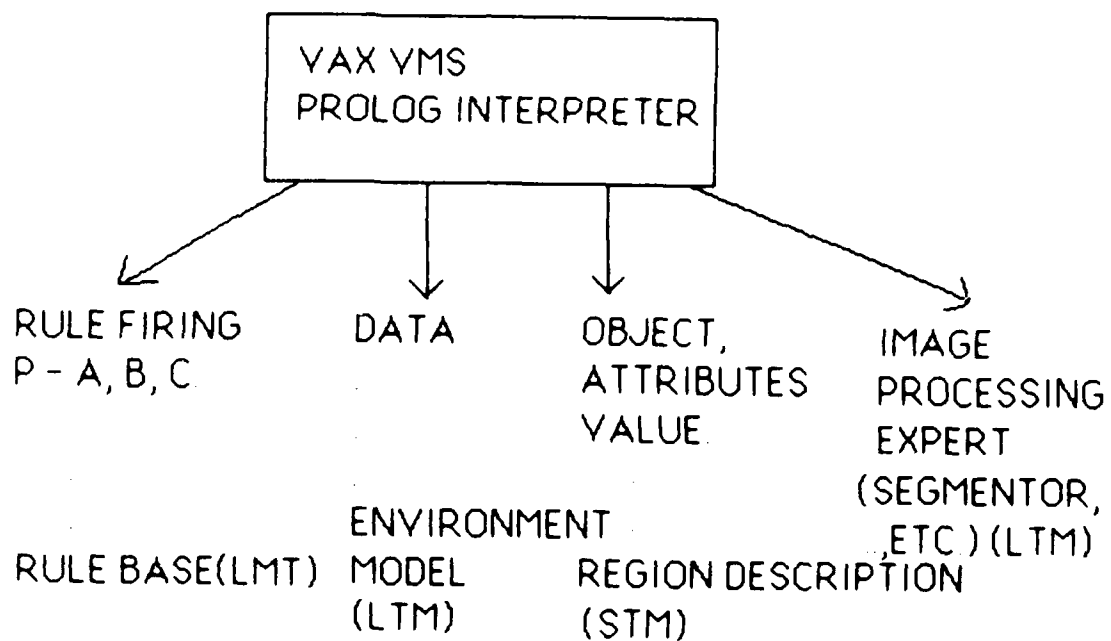
FIG2.3 ENVIRONMENT FOR IMPLEMENTATION

We use a general editor to create and modify the expert system. A PROLOG interpreter written in PASCAL is used to control the rule instantiation and image routines execution. No consistency checking module and no explanation capability are provided in current stage. The characterstics of using PROLOG as an expert system development tool are discussed by Subrahmanyanm. However, the relationship between PROLOG and other general-purpose language are not addressed. Hence we will discuss how PROLOG interpreter is modified to incorporate the capability of interfacing to the other image processing routines, usually written in general-purpose programming languages. The vision expert system includes knowledge base, rule base, and environment model.

Rule base is further divided into control rules and knowledge rules. They will be discussed in Sec. 3.2 and Sec 3.3. Because of the extendability and flexibility requirements, we adopt the rule-based approach. The environmental model is expressed by a set of hierarchical trees and implemented as facts in PROLOG details will be discussed in Sec. 3.1. Both rule bases and enviornmental models constitute the long-term memory (LTM) of our system; while feature descriptions of the segmented regions constitute part of the short-term memory (STM).

2.3 Method of Verification

The procedure of verification is shown in Fig. 2.4. In general, this is a combination of top-down block-selection and bottom-up evidence-combination processes. Since the goal of this system is to verify the existence of the objects in the given input image, we use the coordinate

IMAGE LOCATION

↓

```
┌─────────────────────┐
│ BLOCK TREE          │
│ SELECTION           │
└─────────────────────┘
```

↓

```
┌─────────────────────┐
│ TREE TRAVERSAL TO   │
│ TERMINAL NODE       │
└─────────────────────┘
```

↓

```
┌─────────────────────┐
│ PROJECTION          │
└─────────────────────┘
```

↓

```
┌─────────────────────┐
│ SELECTING SEGMENTED │
│ REGIONS             │
└─────────────────────┘
```

↓

```
┌─────────────────────┐
│ COMBINING CERTAINTY │
│ FACTOR FOR PROPERTY │
└─────────────────────┘
```

↓

```
┌─────────────────────┐
│ REFINING IF NECESSARY│
│ MODEL GUIDED SPLIT AND│
│ MERGE               │
└─────────────────────┘
```
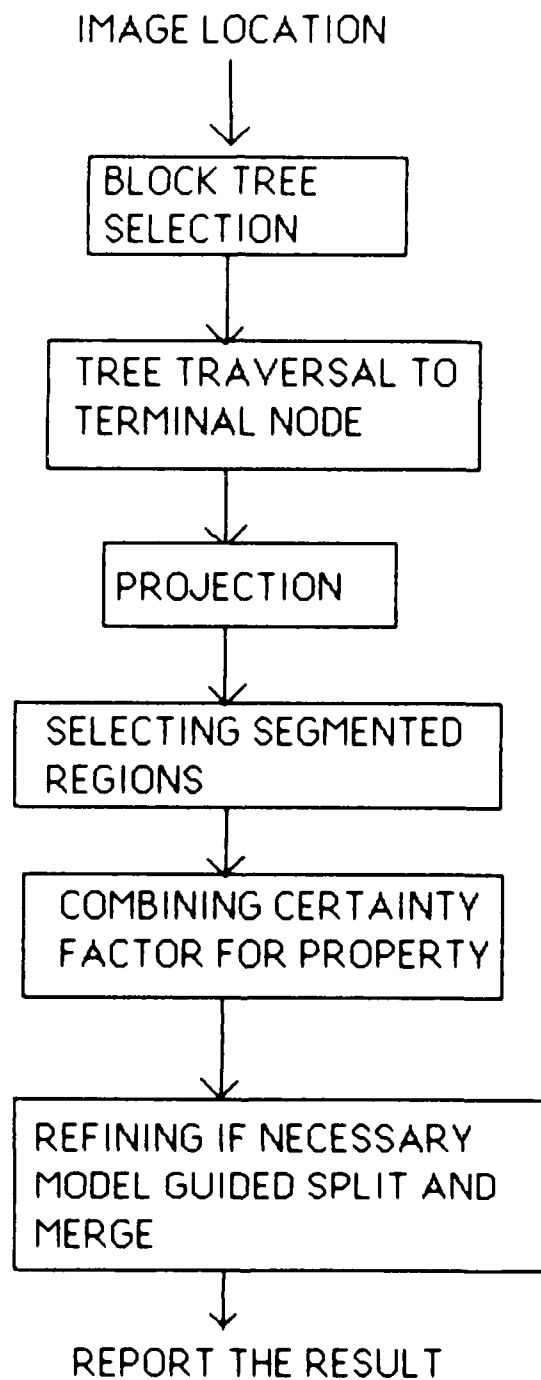
↓

REPORT THE RESULT

FIG 2.4 PROCEDURE FOR VERIFICATION.

of the model to select candidate block or a part of a block for verification. After the block tree discussed in Sec. 3.1, is selected, it is depth-first traversal to the terminal nodes, which represent primitive planar surfaces. The vertices of the boundary were extracted and we project them into the 2-D image plane. Then the projected region is intersected with the segmented image regions. If the area of the intersection area is large enough, the regions are selected for advance checking. The properties being checked are based on the property list in the environmental model. After the certainty value for each property is evaluated, they are combined based on the weighted average sum. The combined value is then justified by a subjective rule. If the certainty value is too low to get a satisfactory unchanged answer, the checked image regions will be refined. This work will be conducted by a model-based split and merge algorithm. Finally, the certainty value will be reevaluated, combined and justified again. The status will be reported to the end-user and used to update its father certainty value.

2.4 Relationship Between Control System and Image Processing Routines

In this section, we will discuss the relationship between the control system and image processing routines. The control system, a set of control rules, written in PROLOG, can initiate image processing routines, written in general purpose language, directly. The relationship between them is shown in Fig. 2.5. The format of control rules which initiates the image processing routine is as follows:

$image'(i_o_routine, out_list, in_vari);

PROLOG CONTROL RULE
'$IMAGE'(I_P_ROUTINE, OUT_LIST,
IN_VARI)

LEVEL 1 — PROLOG INTERPRETER

MAIL BOX

LEVEL 2 — COMMAND DISPATCH

MAIL BOX

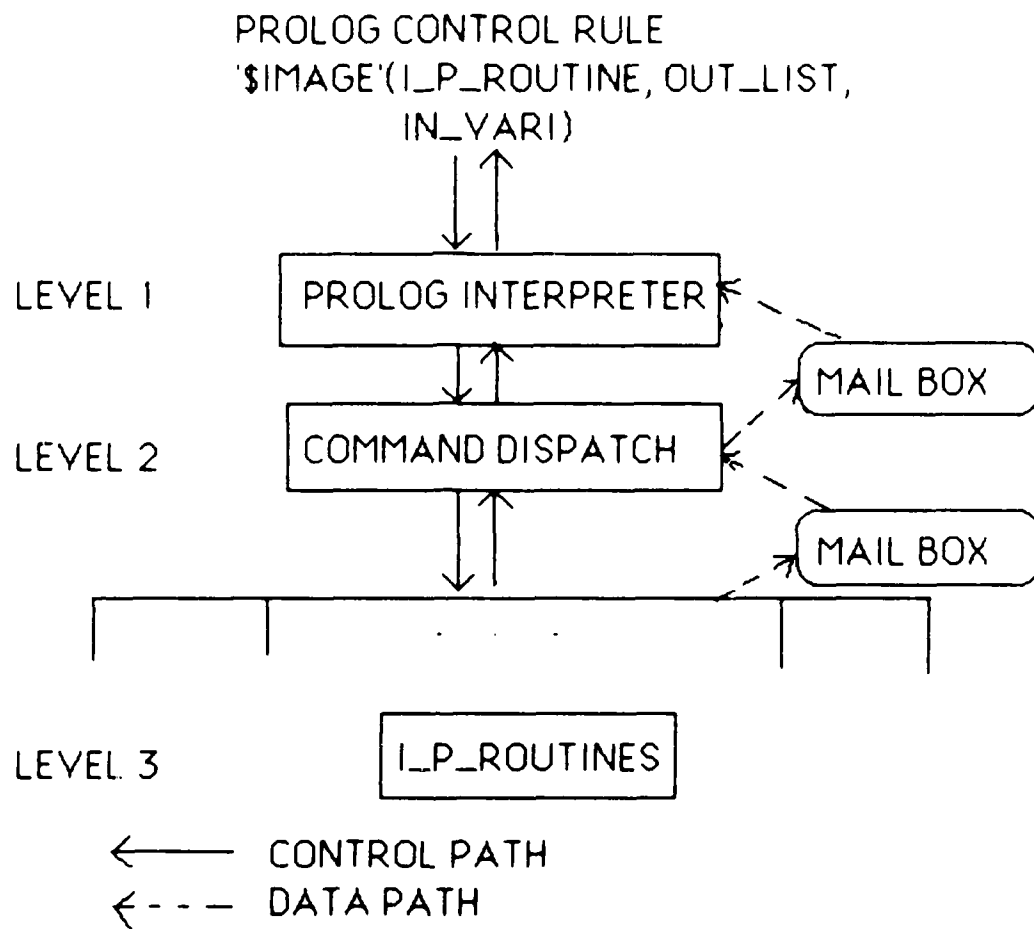LEVEL 3 — I_P_ROUTINES

⟵——— CONTROL PATH
⟵- - - DATA PATH

FIG. 2.5 RELATIONSHIP BETWEEN CONTROL RULE
AND IMAGE PROCESSING ROUTINES.

- 15 -

where

(1) `$image` is a build-in PROLOG predicate, that is, a functor.

(2) i_o_routine is the name of the image-processing routine.

(3) out_list is the list of arguments which are transmitted to i_o_routines.

(4) in_vari is a variable which will unify with a list of arguments received from i_o_routine.

The PROLOG interpreter parses this predicate, builds the necessary data structures and then activates the command dispatcher. The dispatcher, which is an independent process, looks up the name of the image processing routine in a command name table. If the name is found, the corresponding image routine will be activated. Adopting this approach, the system developer can easily incorporate new image processing routines. Once the image processing routine is tested successfully, he need only modify the routine slightly to accept the input parameters from the dispatcher and transmit the output parameters to the dispatcher. Additionally, the name of the new procedure must be added to the command name table in the dispatcher.

In order to synchronize the communication between control rules and image processing routines, we use a mail-box to transmit and receive data. A mail-box is a file-like data structure. It is one of the methods for data communication between different processes. Through a mail-box, i/o operations can be automatically synchronized.

For large quantitative data, the conventional file structure is used to transfer the information. However, the frequently referred data are

put in the blackboard. The details will be shown in Sec. 4.1.

III. KNOWLEDGE BASE

In this section, we shall discuss the structure of the knowledge base. It consists of two parts: an environmental model and a rule base. The rule base can be divided into two subparts which are described in detail below

3.1 Environmental Model

One of the important features in our system is that we assume an exact 3D environmental model is known. We can use every possible method, e.g., measuring directly, getting from documents or maps, to create this model. Here we use stereo images to estimate the 3D information. The related mathematics already has been treated in Sec. 2.1.

An environmental model is a set of block-trees, which consists of a ground area surrounded by several roads. Each block in the system is represented as a hierarchy tree. It is the static data base of our change detection system. Each node has a unique name and a set of attributes. They are represented as a set of facts in PROLOG. The format is 'attribute (node-name, value)'. This is a simple representation scheme. However, the structure is not organized well enough.

The root of a block tree is the block indentification node. It contains the minimal bounding rectangular (m.b.r) of the block. During the varification process, m.b.r. is used to select the target blocks of the

model from an aerial photograph.

Property inheritance is, for example, one of the system features. With this feature, the representation will become more compact. But, the processing will become more complicated. If property 'background-smc' for node 'building1' is the same as its parent node and it is wanted, the system will backtrack in the block tree to find the required property in its ancestor.

The hierarchical relationship of the model is predefined. For example, the building-family is a predefined path in the tree from top to bottom which is shown in Fig. 3.1.

3.2 Control Rules

In this section, we shall describe how the verification procedure is implemented in PROLOG. Because the areas being verified are modelled by a block-tree and the verification process is a depth-first tree traversal, we need an inherently recursive language for good implementation. PROLOG is just the answer. Furthermore, the judgement of the verification result also strongly suggests that we us the logic programming.

/* Control Rule 1 */

If the process is intialization, then

(1) read image-dependent data, including region map, image file name, debugging flag, image-identification, and region description file name,
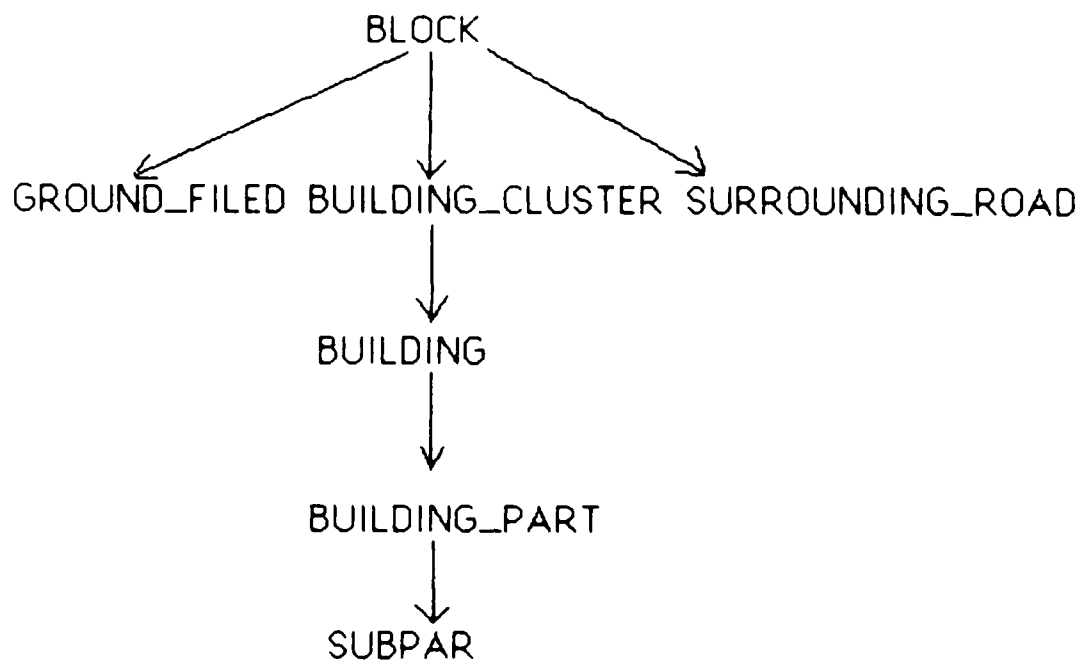
```
                          BLOCK

      GROUND_FILED  BUILDING_CLUSTER  SURROUNDING_ROAD

                        BUILDING

                      BUILDING_PART

                        SUBPAR
```

FIG 3.1 RELATIONSHIP IN THE ENVIRONMENT MODEL

(2) call external FORTRAN routine crbkbd to create a blackboard.

/* Control Rule 2 */

If the process is find-the-reference-ratio, then

(1) get the coordinates and SMC of the reference point,

(2) activate an external routine to calculate reference gray value of the reference point, tref,

(3) get the standard gray value of the reference point's SMC, rref, and

(4) set the ratio tref / rref.

/* Control Rule 3*/

If the intersection area of a 'block' and a segmented image is large enough, then add the block to the checking list.

If the intersection area is too small, then try the next block-tree.

If the intersection area is medium, then check the descendant.

/*Control Rule 4*/

If the select-block process is over, then activate the first node in the checking list.

/* Control Rule 5 */

If Node N is a nonterminal node, then add its children to the checking list.

/* Control Rule 6 */

If Node N is a terminal node, then calculate its certainty value by combining the certainty factor of each property in the property list.

/* Control Rule 7 */

If the certainty factors of all children of a specific node are completely decided, then decide its certainty factor by combining the certainty factor from its own property list and that from its children.

(Note: The combined certainty is the average of the certainty factors of the properties in the property list and that of the children.)

/* Control Rule 8 */

If the certainty factor of a node is decided, then report its status.

/* Control Rule 9 */

If the certainty of a node can not be decided after all properties are evaluated, then try to 'split and merge' the pictures associated with the node.

3.3 Knowledge Rules

The rules used to compute the certainty factor of each property are called knowledge rules. These rules are subjective in nature. End users can improve the system performance by tuning these rules to reflect their own viewpoints. A domain expert can add new knowledge rules to the system without interfering with other rules.

There are several reasons why there are not many rules listed here. First, the authors are not domain experts and the current system is only a prototype system. Secondly, the objects modeled now have only a planar surface and can be approximated by polygons. If complex objects, such as, gasoline tanks, are taken into consideration, then the modelling method must be modified and additional knowledge rules must be created. Thirdly, we consider only verification in the current state. It is rather simpler than the interpretation process. McKeown [4] has shown that most of the knowledge rules used for image interpretation are consistent rules.

3.3 Size Rule

/* Rule 1 — certainty of size */ There are two different rules for size comparison. The first one is based on the area and the second one is based on the polygons. Since we assume that an exact 3D model is given, we perform the judgement just by mapping the 3D objects into 2D image space and then comparing them directly.

The 'certainty of size' is easier to find than the 'certainty of polygon'. However, in the case of verifying the status of a building, the certainty value of the polygon is more reliable than the certainty value of the size. We cope with this situation by putting a higher weight in the property polygon.

The procedure of finding the certainty value of property size can be explained briefly below. First, the regions corresponding to a node is found by an external routine. Secondly, the size of the node, X, is calculated by counting the number of pixels of the projected image of the 3D object. The size of the corresponding regions, Y, are summed together. A certainty value is assigned according to the ratio of X to Y. The real rule in PROLOG is shown below:

/* Rule 2 — certainty of polygon */

The procedure to find the certainty of property polygon is similar to that above, except that the area of the intersection region between the projected node and the corresponding regions is used instead of the total summation size of the corresponding regions.

3.3.2 Gray Level Rule

The average gray value of a specific surface depends on the surface material, the light source, the surface direction, the imaging sensor, and the sensing environment, etc. In most computer vision systems, the absolute gray level is seldom regarded as a useful feature. Instead, people

usually use the region uniformity as an important knowledge source. But, in most situations we know that the gray intensity is strongly correlated to the surface material category. For example, the intensity of a lawn area is darker than the intensity of a gravel area. We can deploy this property and avoid problems by using a relative gray scale.

However, we find that surface direction is an important factor affecting the reflectivity; and unfortunately, our model data is not accurate enough for us to use this knowledge. This inprecision may be improved if we can design a better correspondence solver with subpixel precision.

3.3.3 Shadow Rule

Shadow is an important feature for the object with enough height value. One can use this feature to distinguish two objects with almost the same size and shape, for example, a parking lot and a building. Fig. 3.2 shows the shadow areas formed by a planar surface, ABCDEFGH. Under an overhead camera the shadow areas depend on the location of the light source and the height of the object. In most cases, they are formed as thin regions; the width is only a few pixels. This phenomenon will introduce some extent of difficulty for the detection of the shadow area. We solve this problem by expanding the projected shadow area.

For any planar surface, we need to decide whether a side will form a shadow area or not. For example, side BC in Fig. 3.2 does not produce a shadow area. To determine whether a shadow area is formed, one needs to decide whether the shadow area and the planar surface lie on the same side of the edge. If the answer is positive, that edge can not
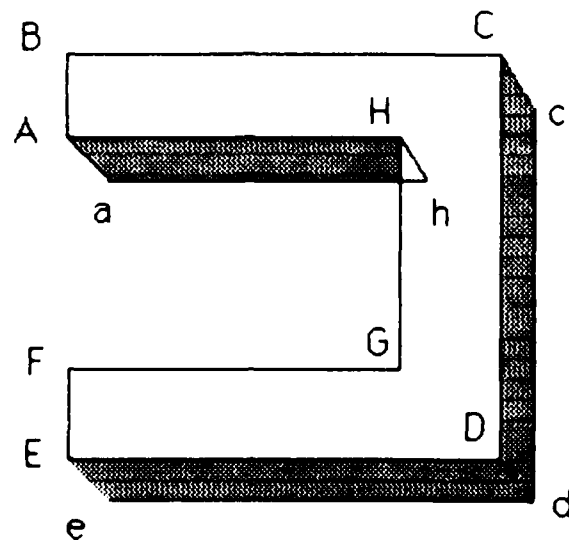
FIG.3.2 SHADOW PRODUCED BY A PLANAR SURFACE
WITH HEIGHT.

form a shadow area. However, if the answer is negative, further checking needs to be performed. Examining AH in Fig. 3.2, we know that the shadow area must exclude the portion under the planar surface.

As explained before, an object is usually decomposed into several planar subregions and a property list is accompanied with the subregion. For two connected Regions A and B, the shadow area of Region A may be occluded by Region B; and vice versa. So the shadow feature cannot be the property of Region A, it must be the property of a node which covers both Regions A and B. The algorithm is shown below:

> Algorithm: Find the certainty value of the shadow feature of a specific node N. Step 1: Find the terminal nodes originated from Node N. Step 2: Find the vertices of the polygon of each terminal node. Step 3: Project the polygon to the image space. Find the intersect regions and their areas. Step 4: Find the sum of the area and the weighted sum of the gray value. Step 5: Assign certainty value to the node according to the difference of gray value between the selected regions and the background.

In our system they are independent to the control system. We will briefly discuss the model-guided split-and-merge algorithm in Sec. 4.

3.4 Certainty Value Combination

There are several methods to combine the evidence. This topic has gained very much attention recently. The methods include Subjective Bayesian theory, uncertainty theory, theory of evidence, possibility the

ory, and fuzzy set theory [8]. In the field of computer vision, especially image interpretation, there is still no consensus. Here, we feel that the weighted average of certainty values is a reasonable choice for our application. Furthermore, research will be conducted in the near future by the authors.

The property list is represented by a fact in PROLOG. Take one example

property(eastwing, [tone, 8, size, 8, polygon, 10])

Here, tone, size, polygon are the properties associated with the node 'east wing'. The numbers 8, 8, and 10 are their respective weights. The last is supplied by the knowledge engineers or domain expert. Another approach is also applicable. That is, we can associate a property list and their weights with a specific item type.

IV. IMAGE PROCESSING UNITS

In a computer vision systems, there must be many image processing tasks. These tasks generally contain a lot of computations.

In the verification subsystem, we segment the image into homogenous regions using the split-and-merge algorithm [9]. For the purpose of verification only, image segmentation is not required. We can project 3D objects into the 2D image space and then compare the features between the projected model and the aerial photograph. However, for the following reasons, we still perform the segmentation process: First, in the near future, we need to gather symbolic measurement for the interpretation

- 27-

process. Second, we want to speed up the processing through symbolic representation. The segmentation and feature extraction process can be performed before the change detection procedure is executed. Third, we can tolerate more errors by considering the features from blocks of pixels instead of pixels.

4.1 Blackboard

In the verification system, some data, as examples, image file, region-map and model-mapped file, are referred to frequently by different image processing routines. Using blackboard I/O, operations are performed efficiently. In the first stage, i.e., the blackboard creation stage, these data are read from secondary storage; in addition, some global variables are initiated. Then all image processing routines can access and modify these data. When the verification process is completed, the blackboard will be copied back to the disk and be deleted from the main storage. This idea sounds good; however, since the space of the blackboard is very big, and the available main storage is very limited, a lot of page faults have occurred during the verification process. In future implementation, we think that those image processing routines should have their own dedicated processing element (processor and memory).

We implement the blackboard in FORTRAN by a global common area. The common block is shown below:

```
Common/board/model, label,
*                     lr, sunang, plncoef, fileseq,
*                     coef, pregs, area,
```

*   peri, compact, m_tone, v_tone,

*   image, mor,

4.2 Model-Guided Split-and-Merge

For any existent modeled node, a model-guided split-and-merge operation will be performed if the certainty value calculation is not conclusive.

The basic idea is to split the regions which are intersected with the projected node region at the intersection boundary. In general, there area large number of small regions generated after the intersection portion is removed. They will be merged with the neighboring regions except the region they are coming from if the average gray values are very similar. The detail algorithm is shown below.

Step 1: Find the intersection region of a segmented region R and the region M, projected from a node of Environmental model.

Step 2: Extract the properties of the intersection region, INTS(R,M).

Step 3: If the area of the intersection region is large enough, then select next region and go to step 1.

Step 4: Assign new region label to the regions formed by extracting a portion of the region from the old segmented region. Measure the properties of the new regions.

Step 5: If the size of the new region is too small, then merge it

with one of the neighboring regions which has the most similar average gray value.

The control program which will initiate the model-guided split and merge is shown below

/* Control Rule 11 */

If the certainty value of a node cannot be decided after all properties are evaluated, then try to 'split and merge' the regions associated with the node.

V. EXPERIMENTAL RESULTS

The goal of this paper is to propose an experimental expert system for change detection. We have already completed the design and testing of the verification subsystem for buildings. The input are 128 x 128 aerial images. Each pixel is represented by an 8-bit gray value.

Fig. 5.1 shows the interpretation process. DEM002 shows, the picture of a building. It is complicated enough. If we can solve the problems related to the building, we should be able to solve all other problems. DEM001 in Fig. 5 shows the region map which is the result of performing Pavlidis split-and-merge algorithm [9]. The regions are relabeled by the model-guided split-and-merge algorithm DEM006 in Fig. 5.1 shows the partial results produced by the control systems. Evidence of each property of each node is shown. From the final results, we know that the building is unchanged.
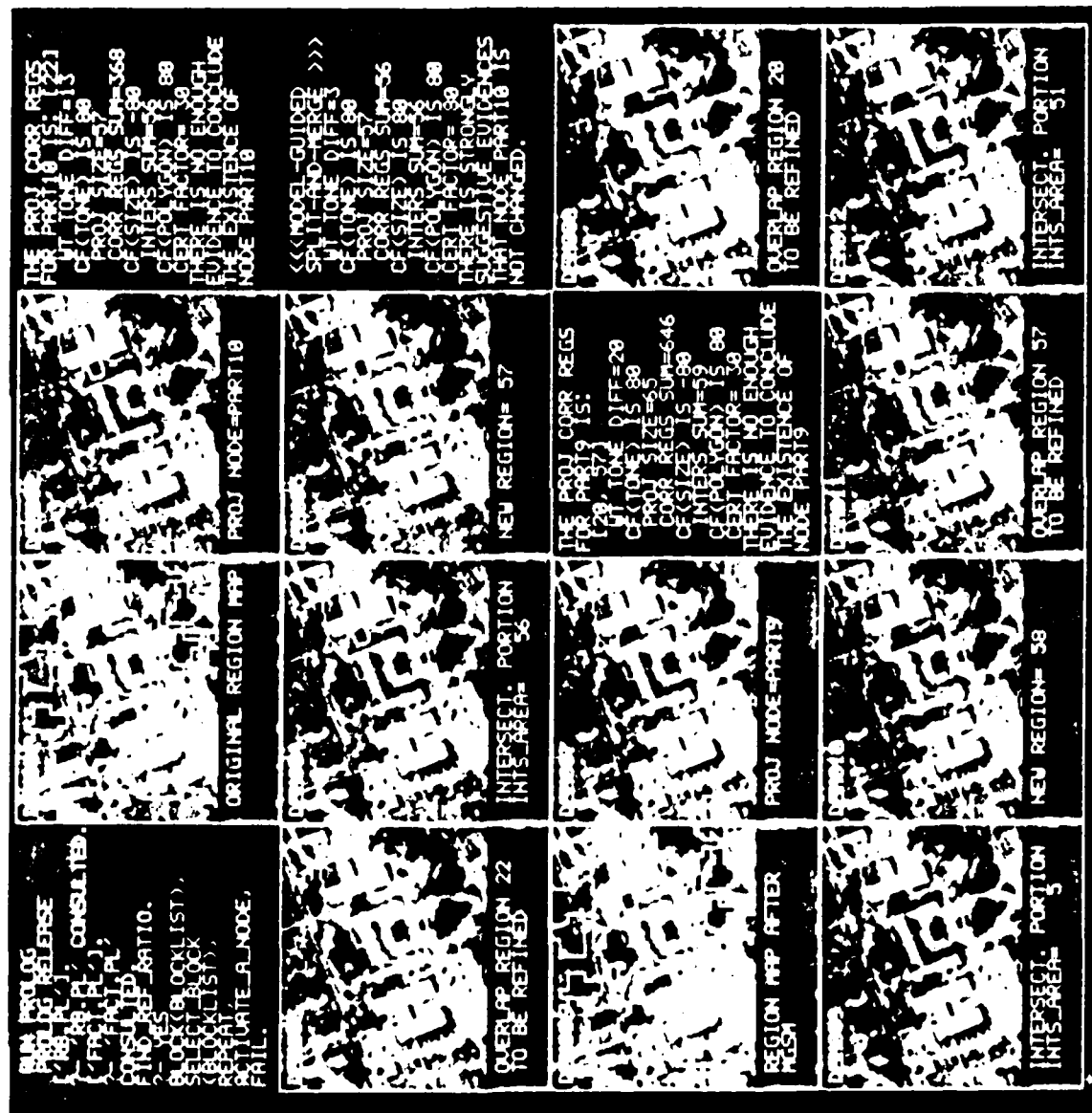
Fig. 5.1 Interpretation Process

-31-

VI. DISCUSSION AND CONCLUSIONS

In this paper, we have proposed an experimental knowledge-based verification system. System organization for change detection is outlined. Knowledge rules and control strategy are described in detail. Designing of the prediction and interpretation subsystems, including a knowledge-based correspondence solving system, is in progress. Currently, we are not satisfied with the performance. However, for the application of change detection, time may not be an important consideration. To improve the system's efficiency, we need to redesign the whole system. A multi-processor system is more suitable. We can use a distributed processing system to perform image processing tasks simultaneously with the control system. If we have a parallel machine, for the computation bound image processing task, we can pursue data parallelism and algorithmic parallelism. In a rule-based system, there are rule-parallelism and evidence paralellism. However, these need a total new expert system designing environment. New tools and new languages are all demanded.

The combination of different evidences is also an interesting problem. Here we use the simplest approach, weighted average value, to cope with this problem. Some computational theories research was conducted in other institutions. What is the most natural one for the application in computer vision system is still a unanswered question.

References

1. R. A. Brooks, "Symbolic Reasoning Among 3-D Models and 2-D Images," Artif. Intell., vol. 17, no. 2, pp. 285-348, 1981.

2. R. A. Brooks, "Model-based 3-D Interpretation of 2-D Images," IEEE Trans. Patern Anal. Mach Intell., vol. PAMI-5, March 1983.

3. A. M. Nazif and M. D. Levine, "Low Level Image Segmentation: An Expert System," IEEE Trans Pattern Anal. Mach. Install., vol. PAMI-6, no. 5, Sept. 1984.

4. D. M. McKeown, W. A. Harvey, Jr., and J. McDermott, "Rule-based Interpretation of Aerial Imagery," IEEE Trans. Pattern Anal. Mach, Intell., vol. 7 no. 5, Sept. 1985.

5. S.A. Barrett, G.J. Kinn, & E. Pfirman, "Automatic Symbolic Change Detection", Final Report, PAR technology Corp, New Hartfor, NY 13413, Aug 1984.

6. M. Tavakoli & A. Rosenfeld, "Toward the Recognition of building & roads on Aerial Photographs, TR-913, University of Maryland, MD. July 1980.

7. K. Price & R. Reddy, "Matching Segments of Images," IEEE Trans PAMI, Vol.1, Jan 1979, pp 110-116.

8. B.G. Buchanan & E.H. Shortliffe, "Rule-Based Expert Systems" Addison-Wesley 1984.

9. S.L. Horowitz & T. Pavlidis, "Picture Segmentation by a directed split-&-merge procedure". 2nd Intermediate Conference on Pattern Recognition p.p 424-433, 1974.

Distribution list

Jack Teller	1
William Alford	1
DMASPOEM	
8301 Greensboro Drive, Suite 800	
McLean, Virginia 22102-3692	
LT Virginia Oard	1
Dr. Richard A. Berg	1
DMAHTC	
U.S. Naval Observatory, Bldg. 56	
Washington, D.C. 20305-3000	
Dan Rusco	1
Defense Mapping Agency Aeronautic Center	
St. Louis AFS, Missouri 63118	
Chin-Hwa Lee	15
Naval Postgraduate School, Code 62Le,	
Monterey, California 93908	
David McKeown	1
CMU	
Department of Computer Science	
Pittsburgh, Pennsylvania 15213	
A. Rosenfield	1
University of Maryland	
Center for Automation	
College Park, Maryland 20742	
G. Reynolds	1
University of Massachusetts	
Computer Information Science Dept	
Amherst, Massachusetts 01002	

Charles A. Harlow   1
Louisiana State University
Department of of Electrical and Computer Engineering
Baton Rouge, Louisiana 70803

Prof. R. Nevatia   1
University of Southern California
Department of Electrical Engineering & Compter Science
Powell Hall, Room 234
Los Angeles, California 90084-0273

W. Ball   1
Washington University
Department of Computer Science
St. Louis, Missouri 63130

Janet Hartman   1
James Fleischer   1
CIA, ORD
Washington, D. C. 20505

Dr. C. Walker   1
Naval Ocean Research & Development
Activity, NSTL, Mississipi 39529

John M. Canning   1
Center of Automation
University of Maryland
College Park, Maryland 20742

Defense Technical Information Center   2
Cameron Station
Alexandria, VA   22217

Dudley Knox Library   2
Code 0142
Naval Postgraduate School
Monterey, CA   93943

36-

Director of Research Administration
Code 012
Naval Postgraduate School
Monterey, CA 93943

1

END

DATE
FILMED

DEC.
1987